CS 4700: Foundations of Artificial Intelligence
Spring 2020
Prelim Prep Questions
Quiz 9

20. Consider a Markov Decision Process defined as follows:
    - There are just two states, A and B.
    - In either state there are just two actions, 1 and 2.
        - In either state if you do action 1 then with 90% probability you wind up in state A and with 10% probability you wind up in state B.
        - In either state if you do action 2 then with 90% probability you wind up in state B and with 10% probability you wind up in state A.
        - For any action that lands you in state A your reward is 1.0, and for any action that lands you in state B your reward is 0.0.
    - $\gamma = 0.5$

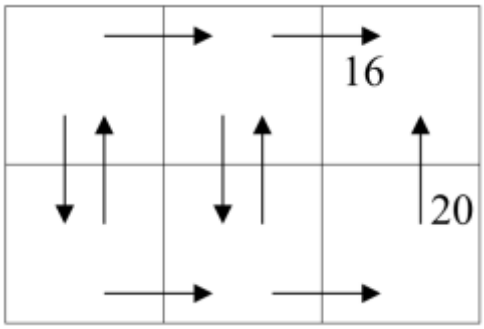
    a. Consider a policy Π that always does action 2 in state A and always does action 1 in state B. What is $U^{\Pi}(A)$?
    b. What is $U^{\Pi}(B)$ for this policy?
    c. What is the optimal policy Π^* for this problem?
    d. What is $U^*(A)$?
    e. What is $U^*(B)$?

21. Consider the same 4-state Markov decision problem used as a running example in class, only where the states "wrap around" if you go off the edge of the graph – going left from <1,1> takes you to <2,1>, going down from <1,1> takes you to <1,2>, and so on. Everything else about the problem is unchanged. Assume γ=0.7.
    a. Policy Π_{UR} says go Up in states <1,1> and <2,1>, and go right in states <2,1> and <2,2>. What is U(<1,1>)?
    b. Do three steps of policy iteration, starting with U_0=0 for all states.

22. There are *N* cities along a major highway that forms a big loop. The cities are numbered 1 through N. If you go clockwise from city *i* you get to city *i*+1 and if you go counter-clockwise you get to city *i*-1, except that counter-clockwise from city 1 puts you in city *N* and clockwise from city N puts you in city 1. You start in city 1. Each day you can either
    - do the STAY action, in which case you'll be in that city the following day
    - do the CLOCKWISE or COUNTER-CLOCKWISE actions, in which case with probability pi you'll wind up in the next city in the selected direction the next day, but with probability (1-pi) your car won't start in which case the city you're in will be unchanged the next day.
2. Entering an even numbered city gives reward $r_i = 1$, and entering an odd numbered city gives reward r_i=0.
    - If for all cities pi = 1 and the discount factor $\gamma = 0.5$:
        a. What is the value of $U^{\pi}(1)$ for the policy π that says to execute STAY in all states?
        b. What is the value of $U^{\pi}(N)$ for this policy?

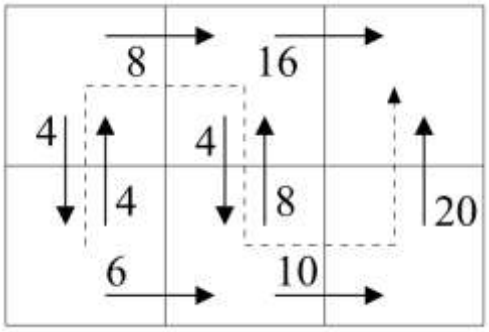
c. What is the optimal value U*(1) for city 1? What policy does it imply?
d. What is the optimal value U*(N) for city N? What policy does it imply?
e. If N=3: Show the values of π for the first two iterations of policy iteration, assuming that for each state π is initially set to CLOCKWISE.

23. Consider the following Markov Decision Process:



An operator that moves to the right is called R, down is D, and up is U. Each action is deterministic. States range from (1,1) at the bottom left to (3,2) at the top right. (To avoid clutter in the figure they are stated here in text rather than on the diagram.) The reward function between any two states is 0 except that R(<3,1>,U,<3,2>) = 20 and R(<2,2>,R,<3,2>) = 16. If there is no arrow corresponding to R, D, or U in a state then the corresponding action does not apply to that state.

Imagine that partway through Q-learning the values of the Q function are as given in the figure below along the actions arrows:



Thus, for example, Q(<1,1>,R) = 6.

Recall that the Q learning update rule is as follows:

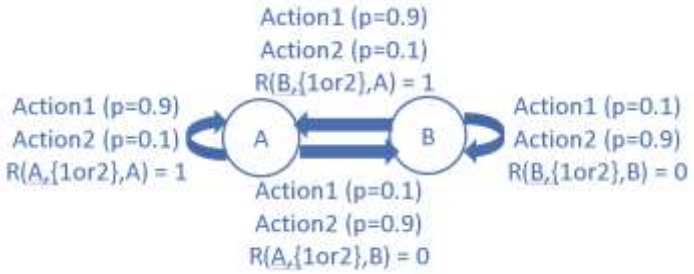
$$\widehat{Q}(s,a) \leftarrow (1-\alpha(N(s,a)))\times \widehat{Q}(s,a) + \alpha(N(s,a))\times(r + \gamma \max_{a' \in A} \widehat{Q}(s', a'))$$

Write on the diagram the new values for each Q if you follow the dashed path (5 actions total). Assume α(n) = 1 and γ = 0.5.

20. Consider a Markov Decision Process defined as follows:
    - There are just two states, A and B.
    - In either state there are just two actions, 1 and 2.
        - In either state if you do action 1 then with 90% probability you wind up in state A and with 10% probability you wind up in state B.
        - In either state if you do action 2 then with 90% probability you wind up in state B and with 10% probability you wind up in state A.
        - For any action that lands you in state A your reward is 1.0, and for any action that lands you in state B your reward is 0.0.
    - γ = 0.5

    a. Consider a policy Π that always does action 2 in state A and always does action 1 in state B. What is $U^{\Pi}(A)$?

This is the MDP depicted in graphical form:



I'll use variable X as shorthand for $U^{\Pi}(A)$ and Y for $U^{\Pi}(B)$.

X = 0.9 x [R(A,2,B) + γ Y + 0.1 x [R(A,2,A) + γ X]
  = 0.9 x [0 + 0.5Y] + 0.1 x [1 + 0.5X] = 0.45Y + 0.1 + 0.05X
0.95X = 0.45Y + 0.1
X = 45Y/95 + 10/95 = 9Y/19 + 2/19

Y = 0.9 x [R(B,1,A) + γ X] + 0.1 x [R(B,1,B) + γ Y]
  = 0.9 x [1 + 0.5X] + 0.1 x [0 + 0.5Y] = 0.9 + 0.45X + 0.05Y
0.95Y = 0.9 + 0.45X
Y = 90/95 + 45X/95 = 18/19 + 9X/19

Plugging in Y into the formula for X gives:

X = 9/19 x [18/19 + 9X/19] + 2/19 = (9x18 + 2x19)/19^2 + (9/19)^2 X = 200/19^2 + (9/19)^2 X
19^2 X = 200 + 9^2 X
(19^2 – 9^2) X = 280A = 200

X = $U^{\Pi}(A)$ = 5/7

    b. What is $U^{\Pi}(B)$ for this policy?

Plugging X=5/7 into the formula for Y gives:

$Y = U^{\Pi}(B) = 18/19 + 9X/19 = (18x7 + 45)/(7x19) = 171/(7x19) = 9/7$

c. What is the optimal policy Π* for this problem?

You want to stay in the state that gives the non-zero reward as much as possible, so you want to keep doing action 1 in both A and B.

d. What is U*(A)?

I'll again use X and Y for the U values for states A and B:

$X = 0.9 \times [R(A,1,A) + \gamma X] + 0.1 \times [R(A,1,B) + \gamma Y]$
$= 0.9 \times [1 + 0.5X] + 0.1 \times [0 + 0.5Y] = 0.9 + 0.45X + 0.05Y$
$0.55X = 0.9 + 0.05Y$
$X = 90/55 + 5Y/55 = 18/11 + Y/11$

$Y = 0.9 \times [R(B,1,A) + \gamma X] + 0.1 \times [R(B,1,B) + \gamma Y]$
$= 0.9 \times [1 + 0.5X] + 0.1 \times [0 + 0.5Y] = 0.9 + 0.45X + 0.05Y$
$0.95Y = 0.9 + 0.45X$
$Y = 90/95 + 45X/95 = 18/19 + 9X/19$

Plugging Y into the formula for X gives:

$X = 18/11 + [18/19 + 9X/19]/11$
$19X = (18x19+18)/11 + 9X/11$
$(19x11)X = (18x19+18) + 9X$
$200X = 360$
$X = 9/5$

e. What is U*(B)?

$Y = 18/19 + 9X/19 = (18x5+9x9)/(5x19) = 171/95 = 9/5$

21. Consider the same 4-state Markov decision problem used as a running example in class, only where the states "wrap around" if you go off the edge of the graph – going left from <1,1> takes you to <2,1>, going down from <1,1> takes you to <1,2>, and so on. Everything else about the problem is unchanged. Assume γ=0.7.
    a. Policy Π_{UR} says go Up in states <1,1> and <2,1>, and go right in states <2,1> and <2,2>. What is U(<1,1>)?
    b. Do three steps of policy iteration, starting with U_0=0 for all states.

(Arbitrarily setting the initial policy to always do D)

	$\hat{U}(11)$	$\hat{U}(12)$	$\hat{U}(21)$	$\hat{U}(22)$	$\hat{\pi}(11)$	$\hat{\pi}(12)$	$\hat{\pi}(21)$	$\hat{\pi}(22)$
0	0	0	0	0	D	D	D	D
1	0.8*(-0.04+ 0.7Û(12)) +0.2*(-1+0.7Û(21)) = -0.232	0.8*(-0.04+0.7Û(11)) +0.2*(1+0.7Û(22)) = 0.168	0.8*(1+0.7Û(22))+ 0.2*(-0.04+0.7Û(11)) = 0.792	0.8*(-1+0.7Û(21)) +0.2*(-0.04+0.7Û(12)) = -0.808	D	L	D	L
2	0.8*(-0.04+ 0.7Û(12)) +0.2*(-1+0.7Û(21)) = 0.8*(-0.04+ 0.7*0.168) +0.2*(-1+0.7*0.792) = -0.027	0.8*(1.0+0.7Û(11))+ 0.2*(-0.04+0.7Û(22)) = 0.8*(1.0+0.7*-0.808)+ 0.2*(-0.04+0.7*-0.232) =0.307	0.8*(1+0.7Û(22))+ 0.2*(-0.04+0.7Û(11)) = 0.8*(1+0.7*-0.808)+ 0.2*(-0.04+0.7*-0.232) = 0.307	0.8*(-0.04+0.7Û(12)) +0.2*(-1+0.7Û(21)) = 0.8*(-0.04+0.7*0.168) +0.2*(-1+0.7*0.792) = -0.027	D	L	D	L
3	0.8*(-0.04+ 0.7Û(12)) +0.2*(-1+0.7Û(21)) = 0.8*(-0.04+ 0.7*0.307) +0.2*(-1+0.7*0.307) = -0.027	0.8*(1.0+0.7Û(11))+ 0.2*(-0.04+0.7Û(22)) = 0.8*(1.0+0.7*-0.027)+ 0.2*(-0.04+0.7*-0.027) =0.773	0.8*(1+0.7Û(22))+ 0.2*(-0.04+0.7Û(11)) = 0.8*(1+0.7*-0.027)+ 0.2*(-0.04+0.7*-0.027) = 0.773	0.8*(-0.04+0.7Û(12)) +0.2*(-1+0.7Û(21)) = 0.8*(-0.04+0.7*0.307) +0.2*(-1+0.7*0.307) = -0.017	D	L	D	L

22. There are *N* cities along a major highway that forms a big loop. The cities are numbered 1 through N. If you go clockwise from city *i* you get to city *i*+1 and if you go counter-clockwise you get to city *i*-1, except that counter-clockwise from city 1 puts you in city *N* and clockwise from city N puts you in city 1. You start in city 1. Each day you can either
    - do the STAY action, in which case you'll be in that city the following day
    - do the CLOCKWISE or COUNTER-CLOCKWISE actions, in which case with probability pi you'll wind up in the next city in the selected direction the next day, but with probability (1-pi) your car won't start in which case the city you're in will be unchanged the next day.

    Entering an even numbered city gives reward $r_i = 1$, and entering an odd numbered city gives reward $r_i=0$.
    - If for all cities pi = 1 and the discount factor γ = 0.5:
        a. What is the value of $U^{\pi}(1)$ for the policy π that says to execute STAY in all states?

$$\sum_{t=0}^{\infty}[\gamma^t \times reward] = \sum_{t=0}^{\infty}[½^t \times 0] = 0$$

        b. What is the value of $U^{\pi}(N)$ for this policy?
        If N is odd, 0, the same as for state 1.
        If N is even, the calculation is as follows.

$$\sum_{t=0}^{\infty}[\gamma^t \times reward] = \sum_{t=0}^{\infty}[½^t \times 1] = 2$$

        c. What is the optimal value U*(1) for city 1? What policy does it imply?
        2.0. The policy would say to go CLOCKWISE, putting you in state 2, with value U*(2)=2. (The optimal policy for all even-numbered states is to STAY, giving the U* value of 2, as in part 2 above.) You then get

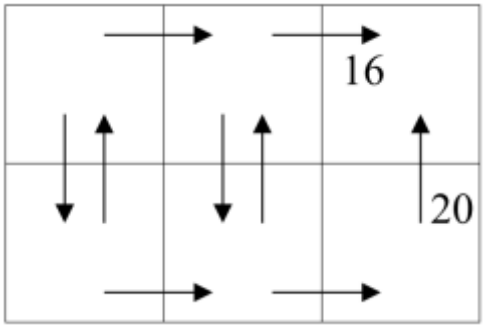
$$1.0 + \gamma \sum_{s' \in [1:N]} P(s'|1, CLOCKWISE)U^*(s')$$
$$= 1.0 + 0.5 \times U^*(2) = 2.0$$

d. What is the optimal value U*(N) for city N? What policy does it imply?
2.0.

e. If N=3: Show the values of π for the first two iterations of policy iteration, assuming that for each state π is initially set to CLOCKWISE.
I give three iterations in case it's of value.

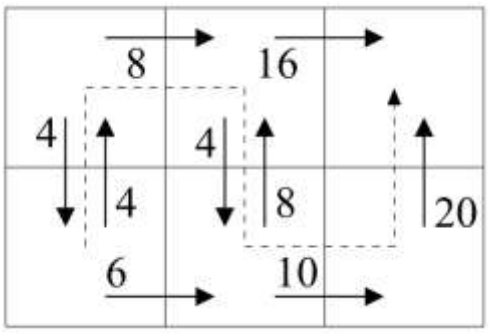
Iteration	π(1)	π(2)	π(3)	U(1)	U(2)	U(3)
Initial Values	CLOCKWISE	CLOCKWISE	CLOCKWISE	0	0	0
1	CLOCKWISE	STAY	COUNTER-CLOCKWISE	1	0	0
2	CLOCKWISE	STAY	COUNTER-CLOCKWISE	1	1	1
3	CLOCKWISE	STAY	COUNTER-CLOCKWISE	1.5	1.5	1.5

23. Consider the following Markov Decision Process:



An operator that moves to the right is called R, down is D, and up is U. Each action is deterministic. States range from (1,1) at the bottom left to (3,2) at the top right. (To avoid clutter in the figure they are stated here in text rather than on the diagram.) The reward function between any two states is 0 except that R(<3,1>,U,<3,2>) = 20 and R(<2,2>,R,<3,2>) = 16. If there is no arrow corresponding to R, D, or U in a state then the corresponding action does not apply to that state.

Imagine that partway through Q-learning the values of the Q function are as given in the figure below along the actions arrows:



Thus, for example, Q(<1,1>,R) = 6.

Recall that the Q learning update rule is as follows:

$$\hat{Q}(s,a) \leftarrow (1- \alpha(N(s,a)))\times \hat{Q}(s,a) + \alpha(N(s,a))\times(r + \gamma \max_{a' \in A} \hat{Q}(s', a'))$$

Write on the diagram the new values for each Q if you follow the dashed path (5 actions total). Assume α(n) = 1 and γ = 0.5.

If α(n) = 1 and γ = 0.5 the update rule becomes:

$$\hat{Q}(s,a) \leftarrow (1 - 1)\times \hat{Q}(s,a) + 1\times(r + 0.5 \max_{a' \in A} \hat{Q}(s', a')) = r + 0.5 \max_{a' \in A} \hat{Q}(s', a')$$

Further, in all but the last move r=0, so you end up replacing $\hat{Q}(s,a)$ in the relevant states with $0.5 \max_{a' \in A} \hat{Q}(s', a')$. Finally, if we assume that we get no further rewards once we reach the end state (that it's a terminal state and all actions keep you there and give reward 0) you simply get 20.